DIGITAL SIGNAL PROCESSING LABORATORY

EXPERIMENT-2 - DESIGNING LOW PASS FILTERS BY WINDOWING METHOD

Group:- 32

Authors:-

Jaya Kishnani (20EC30020)

Gunjan Shekhar(20EC10032)

TA:- Abhishek Singh

Aim:

a) Design FIR filters
b) Observe the response of FIR filters when noise is added to the signal

Theory:

The frequency response of an ideal filter looks like-

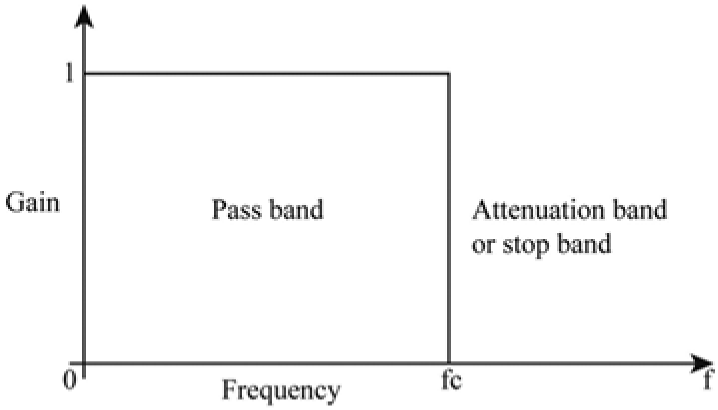


Figure 1

However for achieving this ideal response, the response of the filter will extend up to infinity. So for practical purposes, the filter is truncated using a window function. This window function can then be modified to get the appropriate results.

If hd(n) is the impulse response of the filter and w(n) is the window function, we get-
h(n) = hd(n)*w(n)
where h(n) is the truncated impulse response of the filter obtained as a result of element-wise multiplication of hd(n) and w(n).

In the frequency domain, the frequency response of the truncated filter is the convolution of the frequency domain representations of hd(n) and w(n).

Observation:

Code-

```
N = 8; %No. of samples
k = (N-1)/2;
wc = pi/6; %cut-off frequency
n = 1 : 1: 10*N;
hd = zeros(1,N); %define filter dimension
w1 = zeros(1,N); %define window dimension
w2 = zeros(1,N);
w3 = zeros(1,N);
w4 = zeros(1,N);
w5 = zeros(1,N);

%Low pass filter
for i=1:N;
   if i == k;
      hd(i) = wc/pi;
   else
      hd(i)=(sin(wc*(i-k))/(pi*(i-k)));
   end
end
plot(hd)
xlim([1 N])
title('Low pass filter(N = 8)')


% rectangular window
for i=1:N
   w1(i)=1;
end
figure(1);
subplot(3,2,1);
plot(w1);
xlabel('n')
ylabel('w(n)')
xlim([1 N])
title('rectangular window')


% triangular window
for i=1:N
   w2(i)=1-abs(2*((i-(N-1)/2)/(N-1)));
```

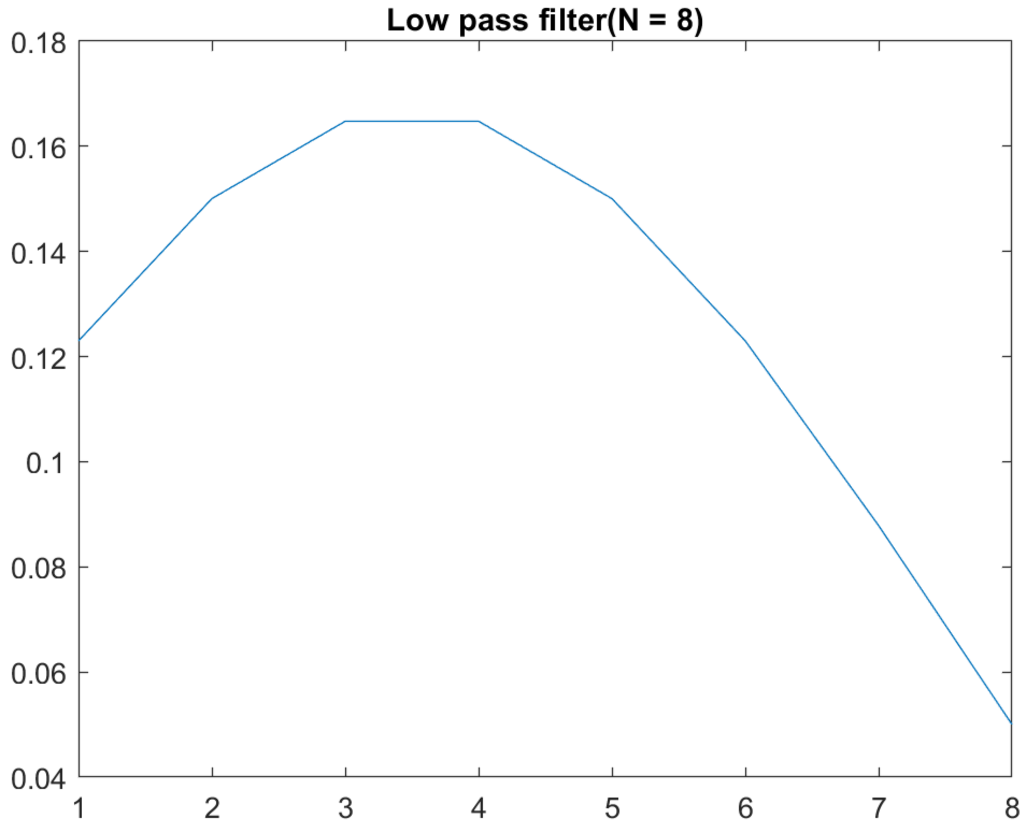
```
end
subplot(3,2,2);
plot(w2);
xlabel('n')
ylabel('w(n)')
xlim([1 N])
title('triangular window')


% hanning window
for i=1:N
   w3(i)=0.5-0.5*cos((2*pi*i)/(N-1));
end
subplot(3,2,3);
plot(w3);
xlabel('n')
ylabel('w(n)')
xlim([1 N])
title('hanning window')


% hamming window
for i=1:N
   w4(i)=0.54-0.46*cos((2*pi*i)/(N-1));
end
subplot(3,2,4);
plot(w4);
xlabel('n')
ylabel('w(n)')
xlim([1 N])
title('hamming window')


% blackman window
for i=1:N
   w5(i)=0.42-0.5*cos((2*pi*i)/(N-1)) +0.08*cos((4*pi*i)/(N-1));
end
subplot(3,2,5);
plot(w5)
xlabel('n')
ylabel('w(n)')
xlim([1 N])
title('blackman window')
```

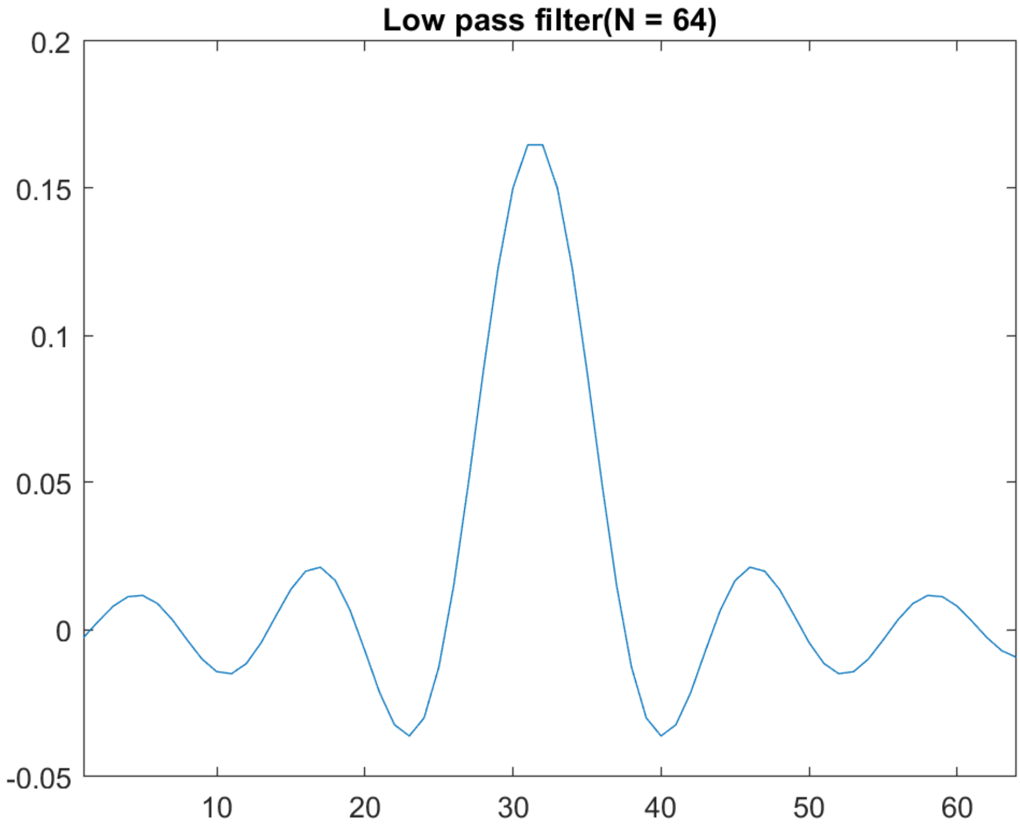
```
sgtitle('Window functions for N = 512')


% Multiplication
figure(2);
f =-pi:0.01:pi;
w = w1; %choosing rectangular window function
h = hd.*w;
freqz(h,1,f);

w = w2; %choosing triangular window function
h = hd.*w;
freqz(h,1,f);

w = w3; %choosing hamming window function
h = hd.*w;
freqz(h,1,f);

w = w4; %choosing hanning window function
h = hd.*w;
freqz(h,1,f);

w = w5; %choosing blackman window function
h = hd.*w;
freqz(h,1,f);

w = w1;
figure;
x = sin((wc/2)*n) + 0.5*sin((wc/4)*n) + 0.5*sin(2*wc*n);%input signal
subplot(4,2,1);
plot(n,x)
title('Input signal')
xlim([0 80]);
X = fft(x); %fft of input signal
X = fftshift(X);
subplot(4,2,2);
plot(abs(X))
title('FFT of input signal')

y = filtfilt(h,1, x); %Filtering input signal
subplot(4,2,3);
plot(y)
xlim([0 80])
title('Filtered Input signal')
```

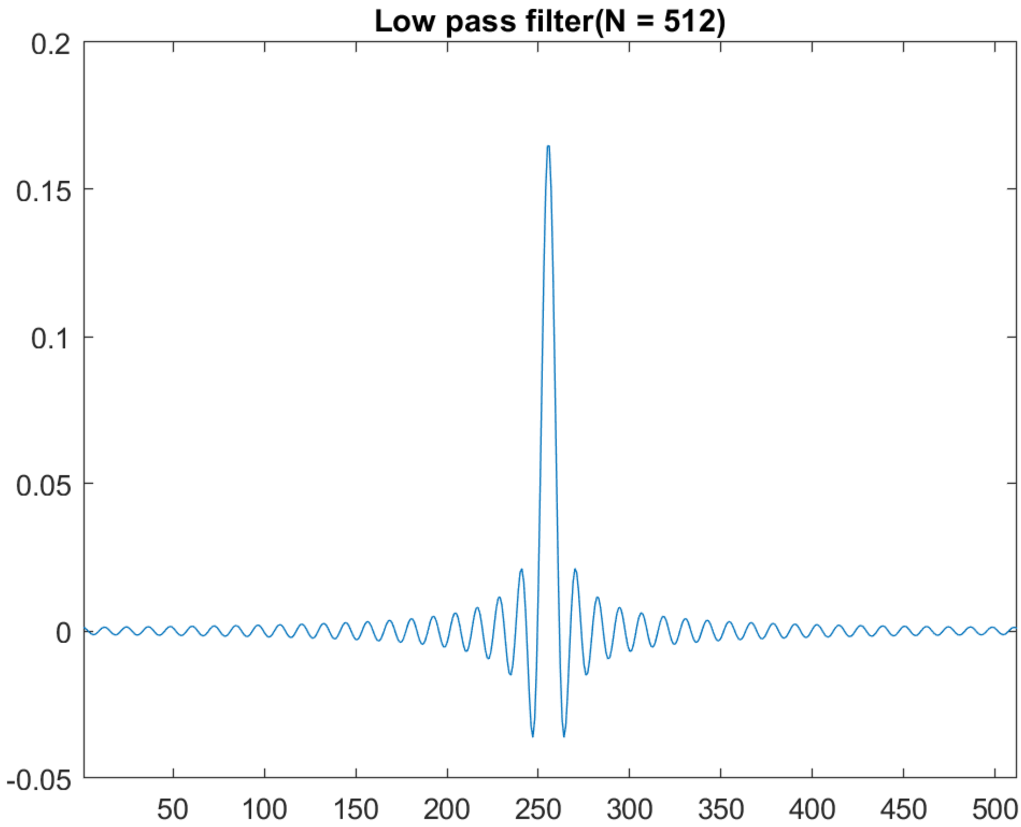
```
Y = fft(y); %FFT of filtered signal
Y = fftshift(Y);
subplot(4,2,4);
plot(abs(Y))
title('FFT of filtered input signal')

noise = rand(size(n)); %noise signal
x = x +noise; %adding noise to input signal
subplot(4,2,5);
plot(n,x)
title('Noisy input signal')
xlim([0 80]);

X = fft(x); %fft of noisy signal
X = fftshift(X);
subplot(4,2,6);
plot(abs(X))
title('FFT of noisy signal')

y = filtfilt(h,1,x);%filtering noisy signal
subplot(4,2,7);
plot(y)
xlim([0 80])
title('Filtered noisy signal')

Y = fft(y); %FFT of filtered noisy signal
Y = fftshift(Y);
subplot(4,2,8);
plot(abs(Y))
title('FFT of filtered noisy signal')
sgtitle("Blackman Window(N=8)")

signal_amp = max(abs(y))
noise_amp = max(abs(noise))
r = snr(x,noise)
```

Plots

1. <u>Low Pass Filter of different orders-</u>

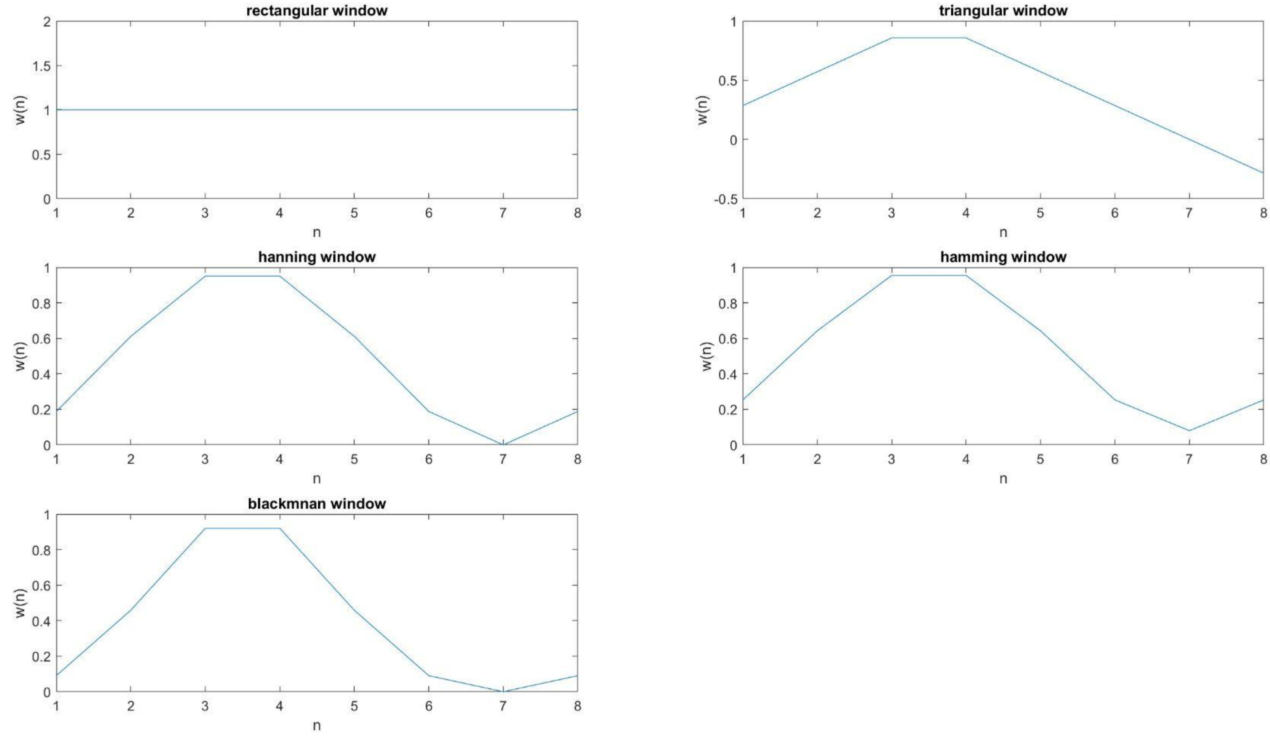



Low pass filter(N = 64)
0.2
0.15
0.1
0.05
0
-0.05
10
20
30
40
50
60

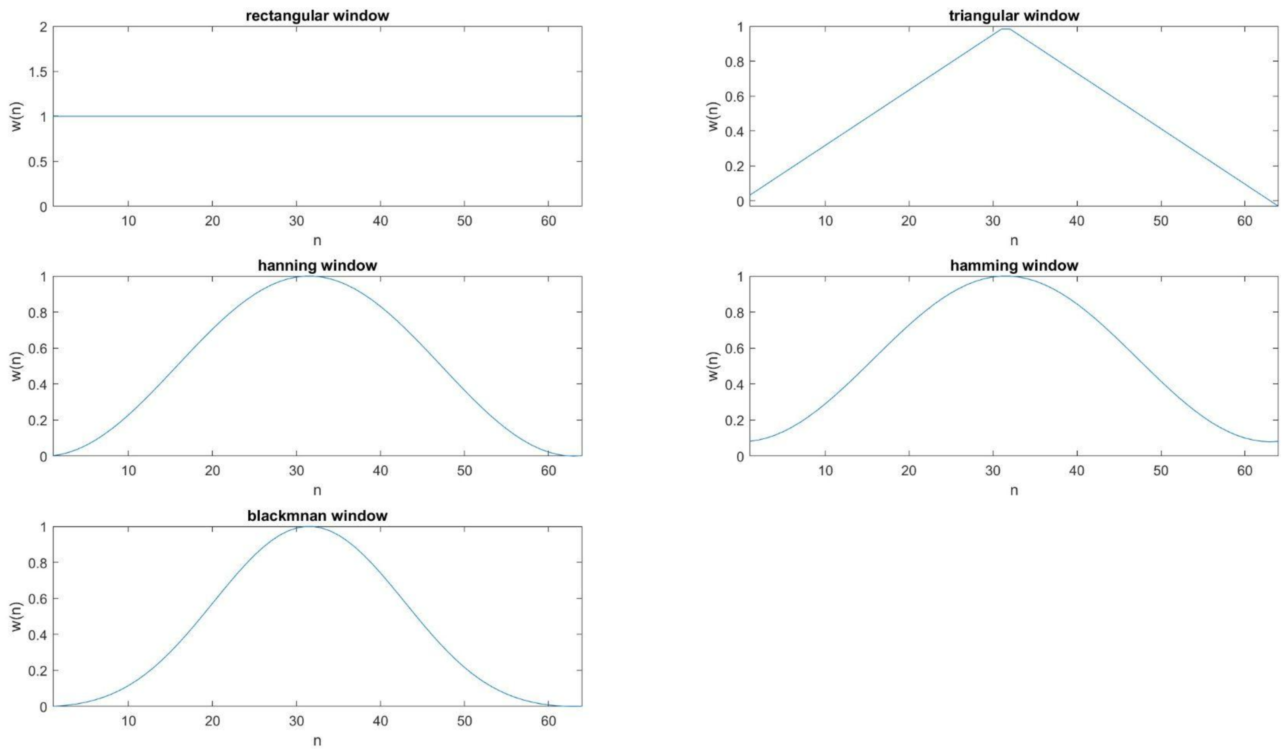

Low pass filter(N = 512)
0.2
0.15
0.1
0.05
0
-0.05
50
100
150
200
250
300
350
400
450
500

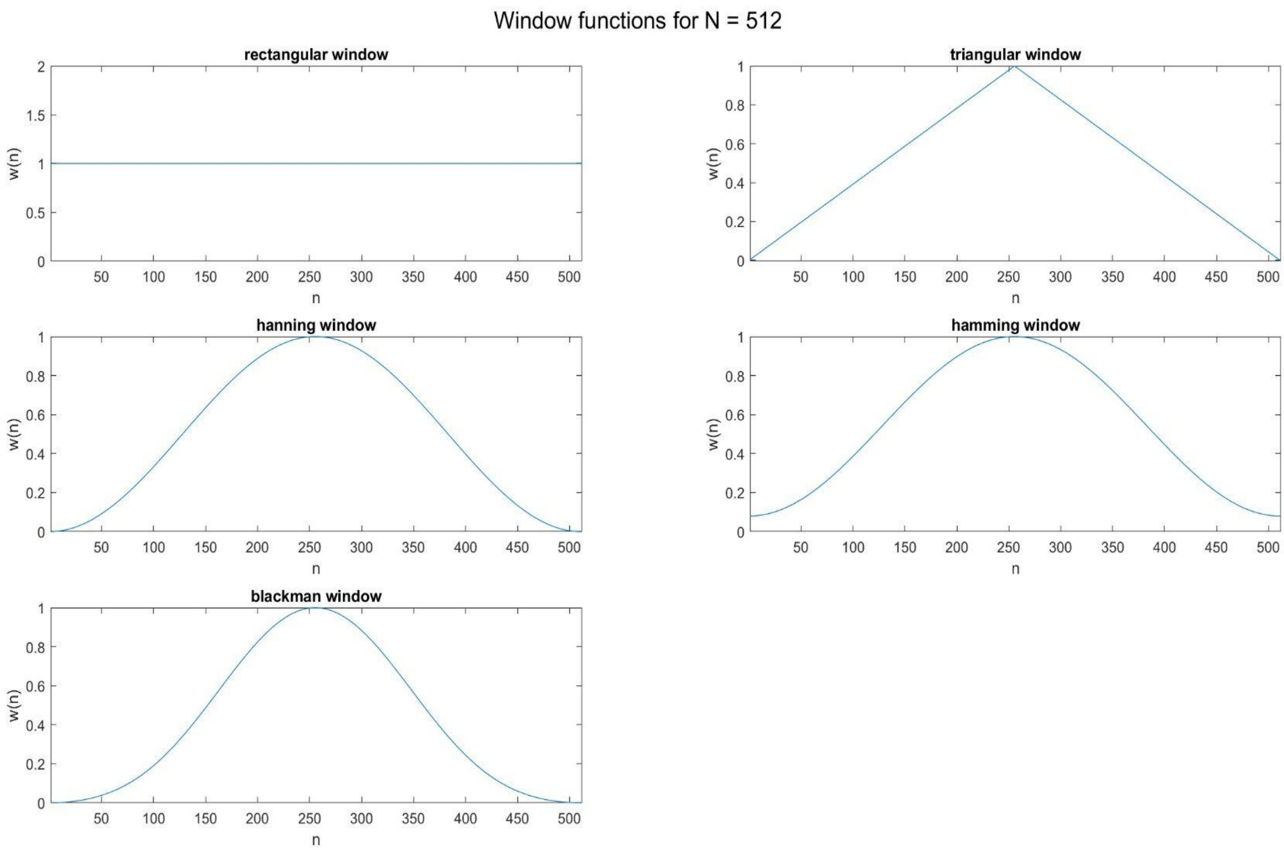
2. Time Domain Representation of Window Signals-

Window functions for N = 8



Window functions for N = 64



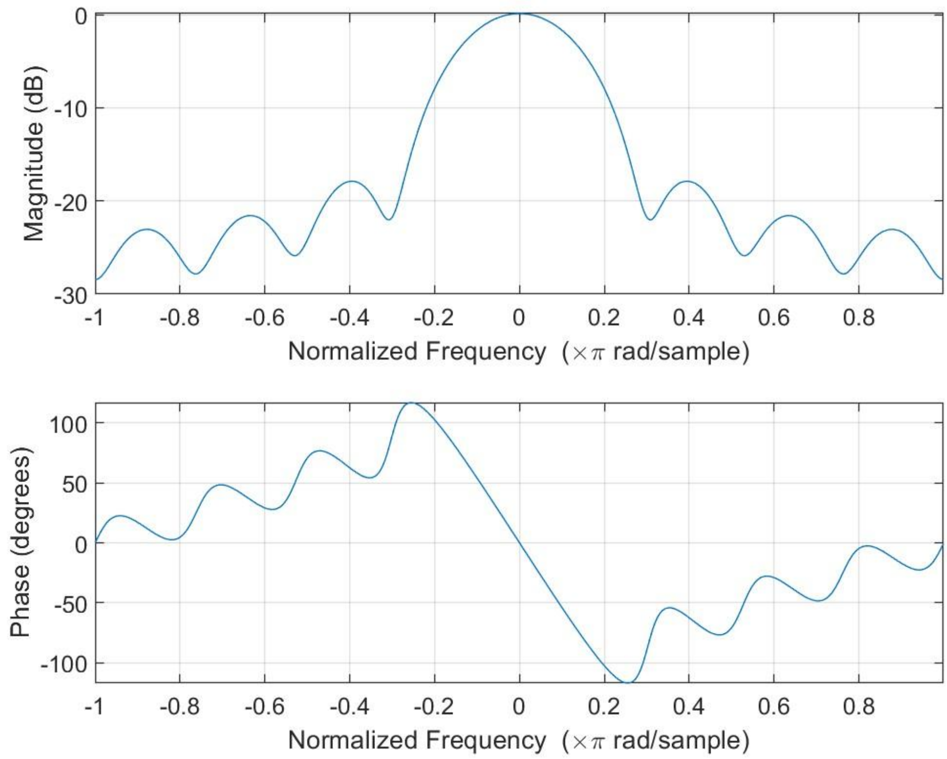


Window functions for N = 512
rectangular window
triangular window
hanning window
hamming window
blackman window
w(n)
n

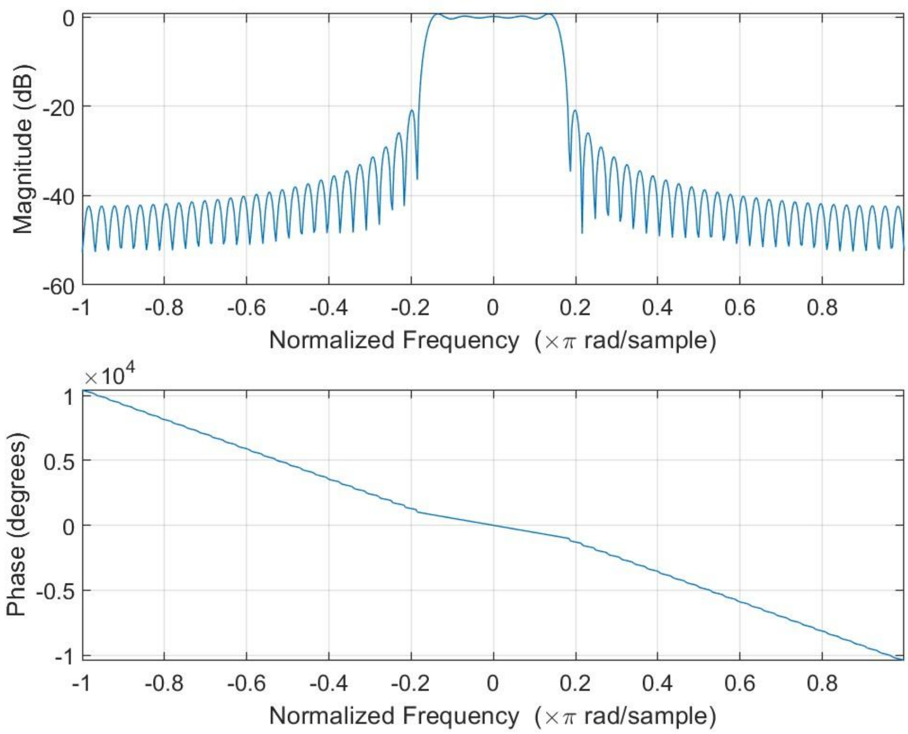
3. <u>Frequency Domain Representation of Window Signals</u>-

a) Rectangular Window-

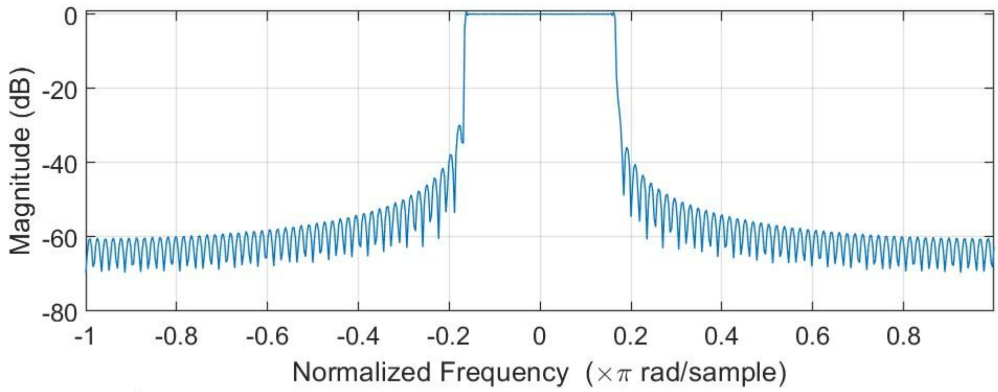
N = 8

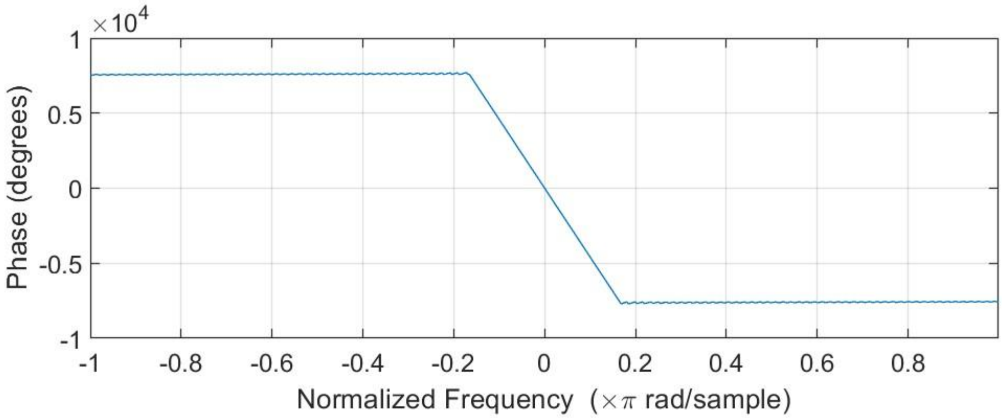


N = 64

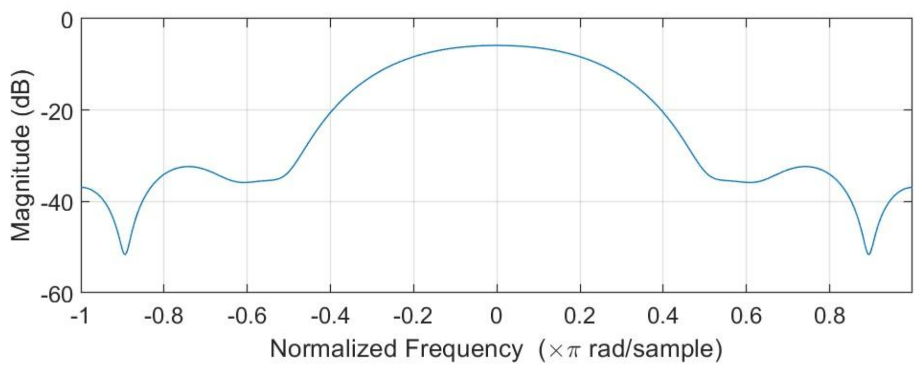


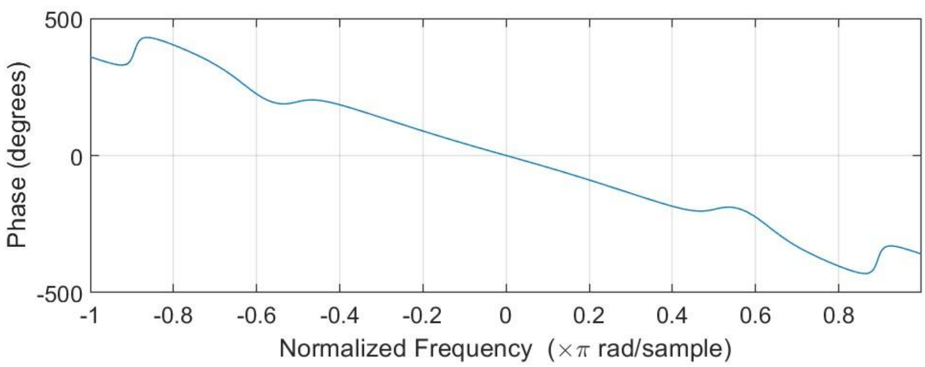
N = 512



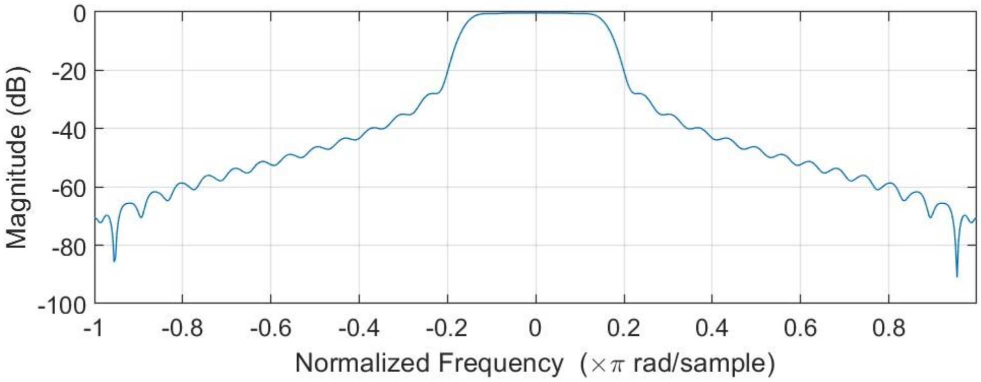


b) Triangular WIndow-
N = 8

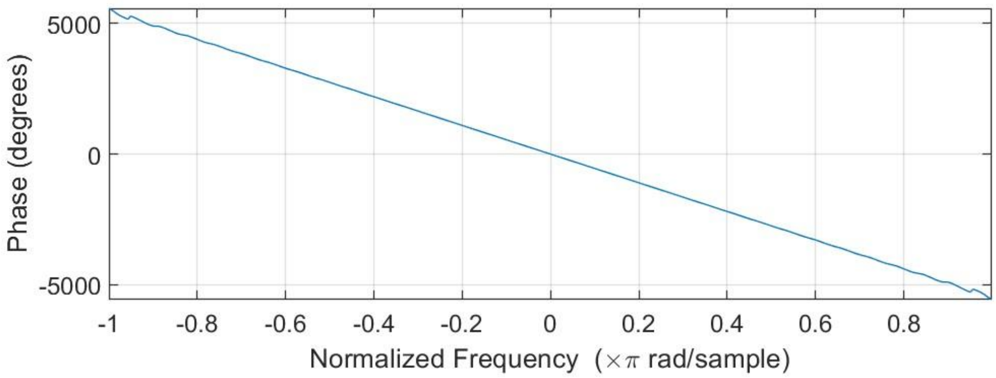




N = 64

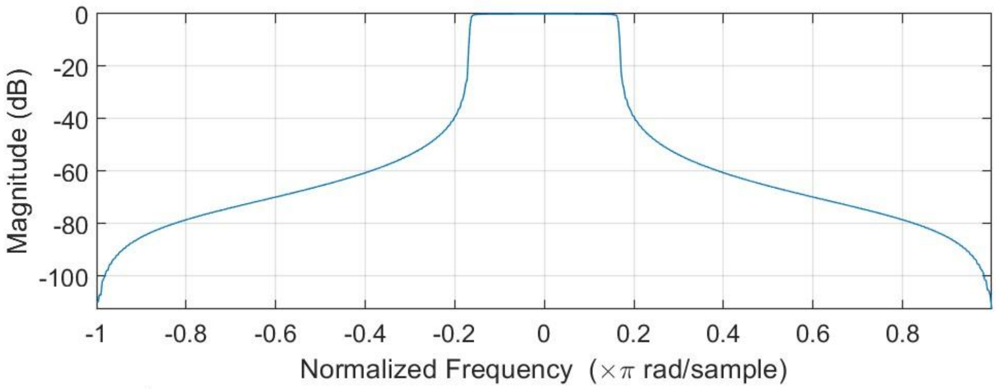


0
-20
-40
-60
-80
-100
Magnitude (dB)
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
Normalized Frequency (×π rad/sample)

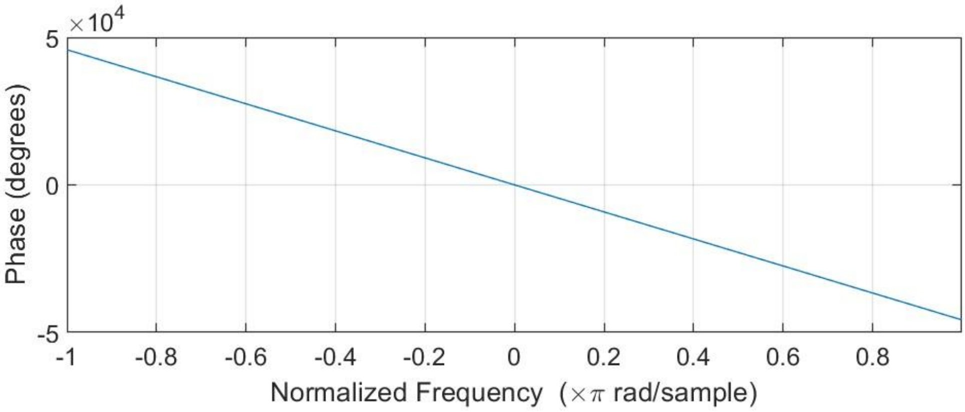


5000
0
-5000
Phase (degrees)
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
Normalized Frequency (×π rad/sample)

N = 512

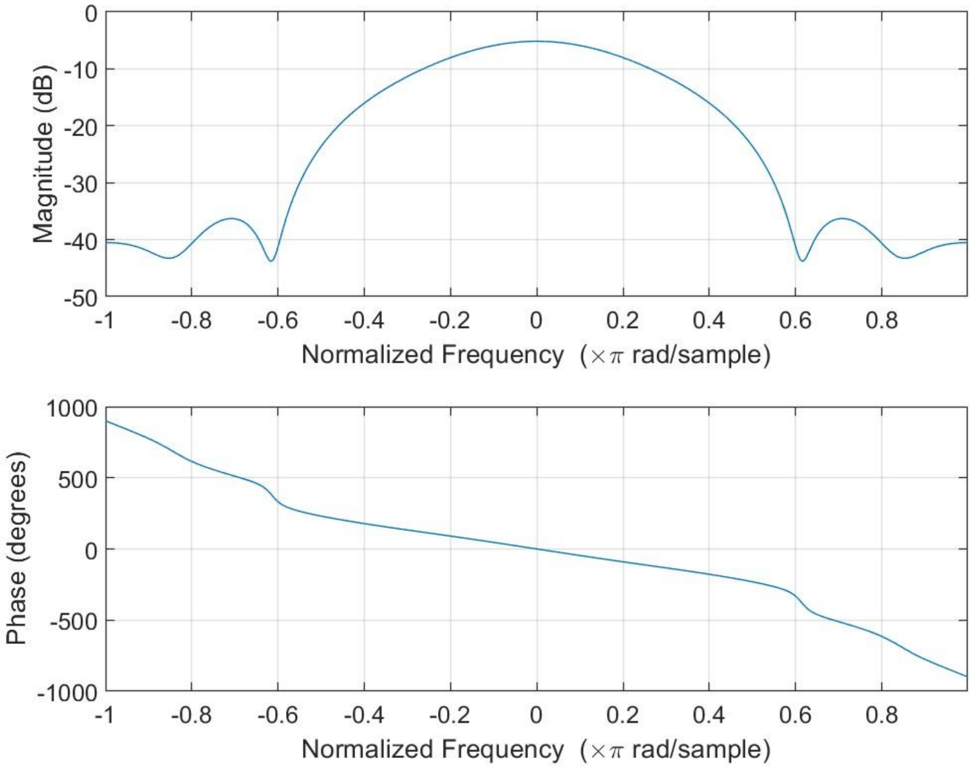


0
-20
-40
-60
-80
-100
Magnitude (dB)
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
Normalized Frequency (×π rad/sample)

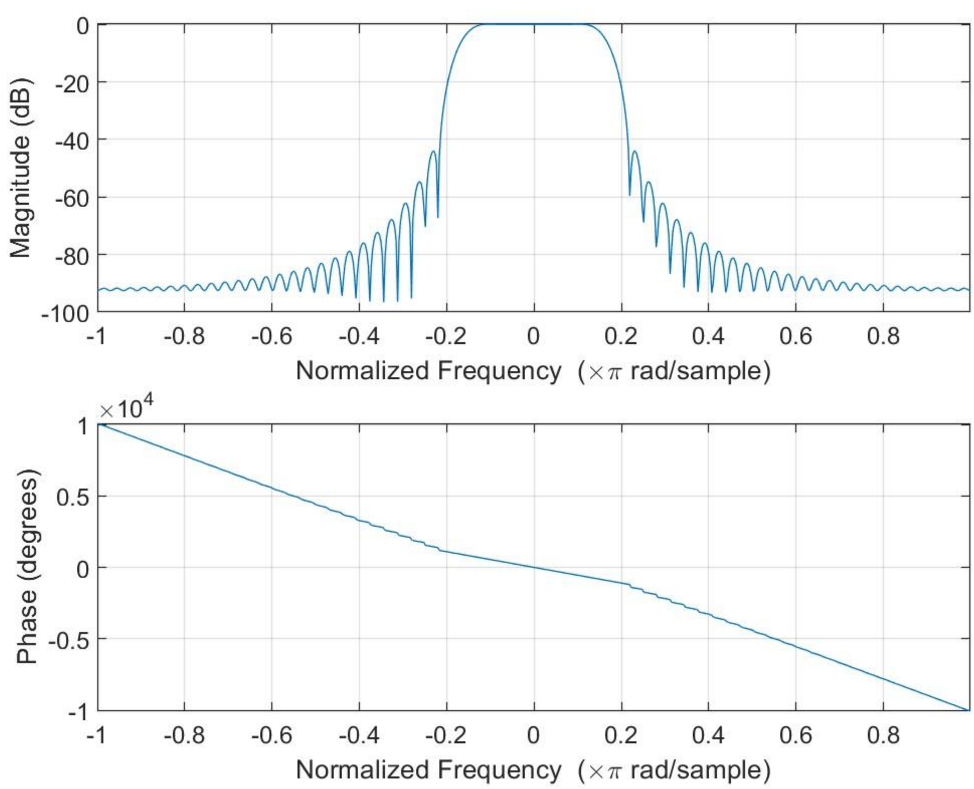


×10^4
5
0
-5
Phase (degrees)
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
Normalized Frequency (×π rad/sample)

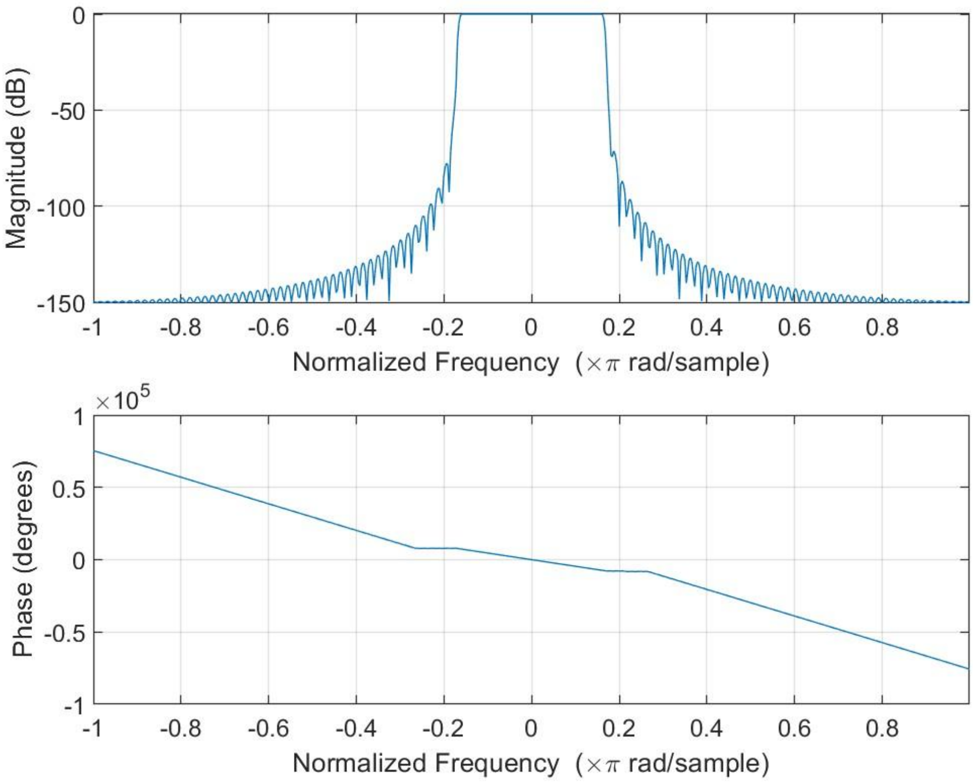
c) Hamming Window-
N = 8



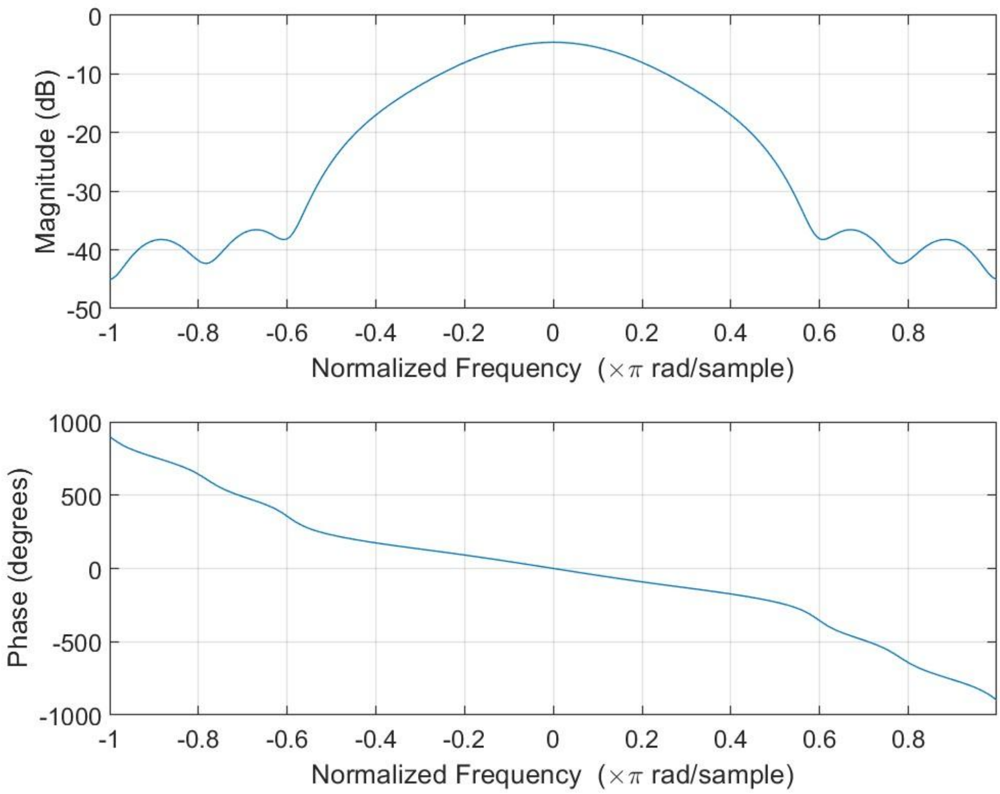
N = 64



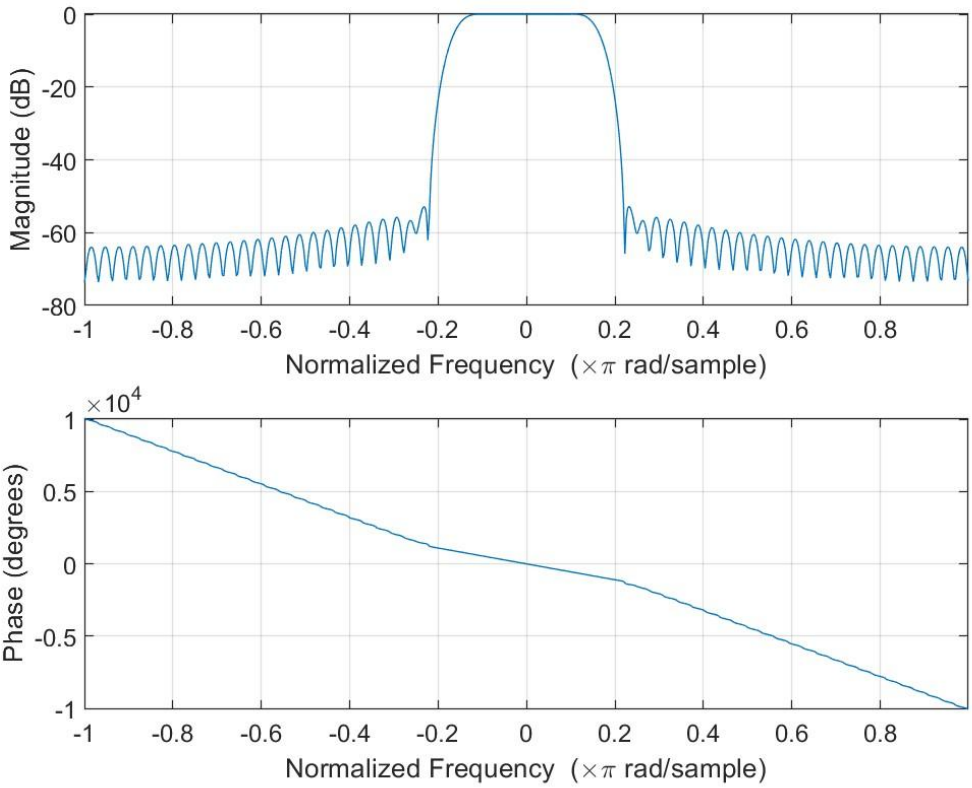
N = 512



d) Hanning Window-
N = 8

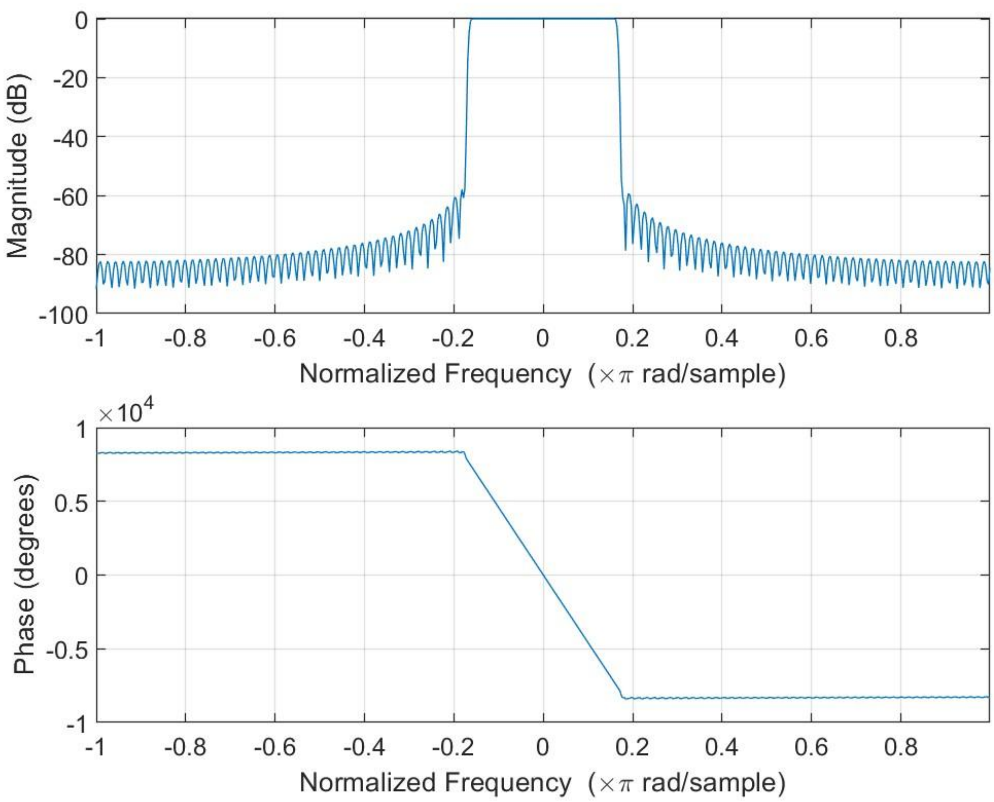


N = 64



0
-20
-40
-60
-80
Magnitude (dB)
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
Normalized Frequency (×π rad/sample)
×10^4
1
0.5
0
-0.5
-1
Phase (degrees)
Normalized Frequency (×π rad/sample)

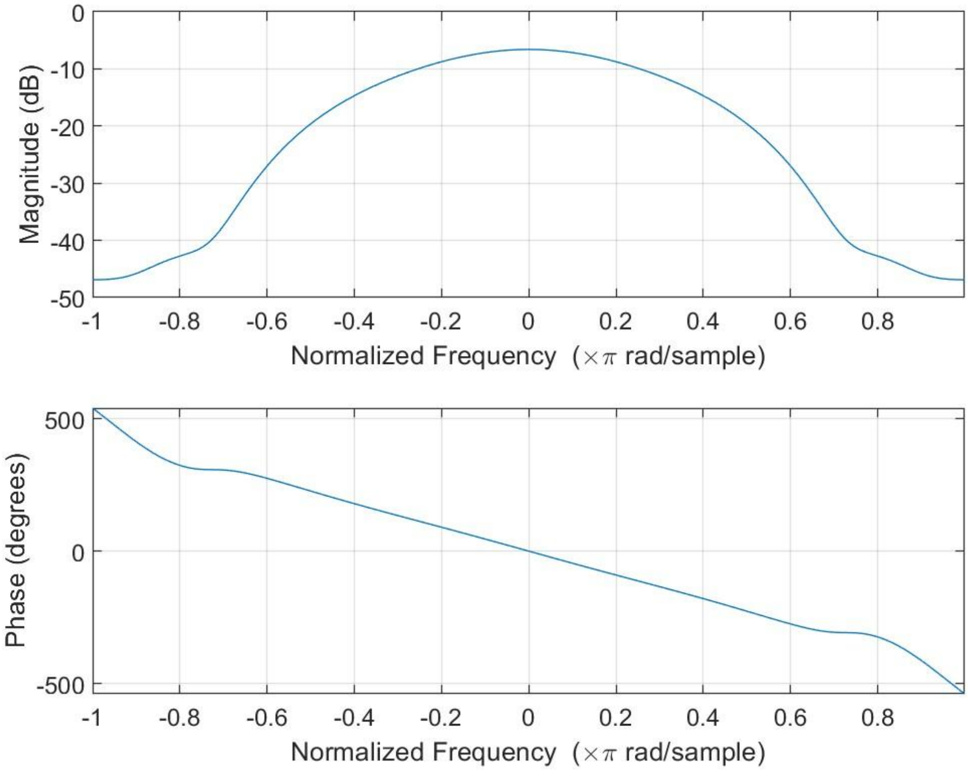
N = 512



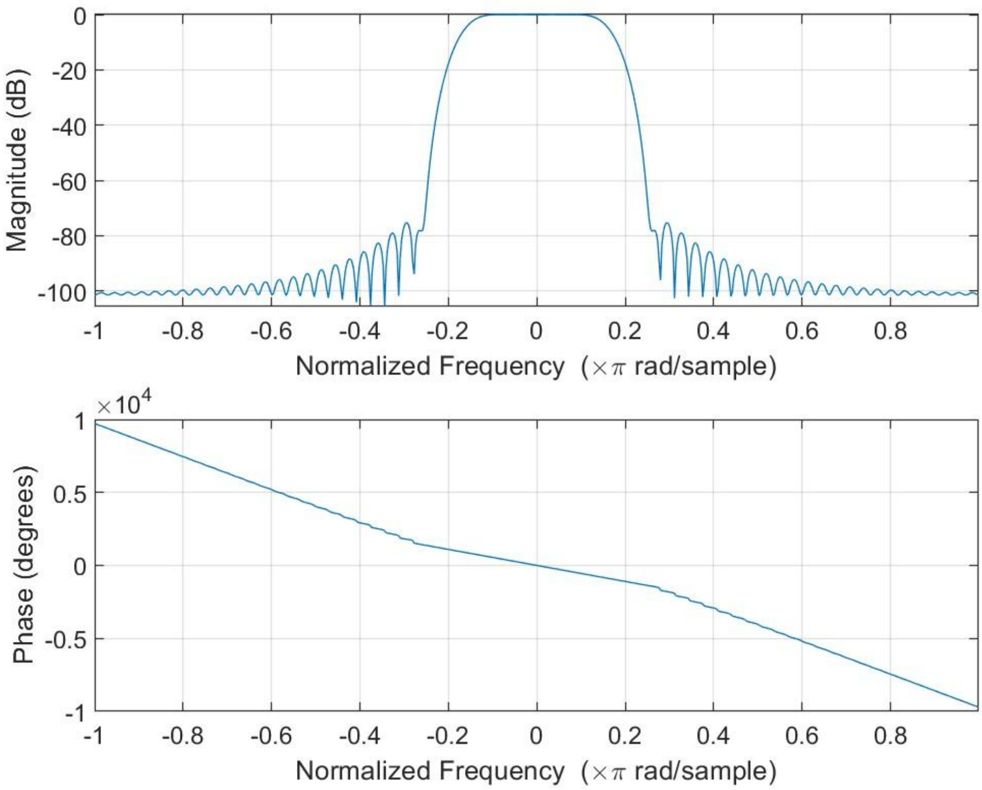
0
-20
-40
-60
-80
-100
Magnitude (dB)
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
Normalized Frequency (×π rad/sample)
×10^4
1
0.5
0
-0.5
-1
Phase (degrees)
Normalized Frequency (×π rad/sample)

e) Blackman Window-

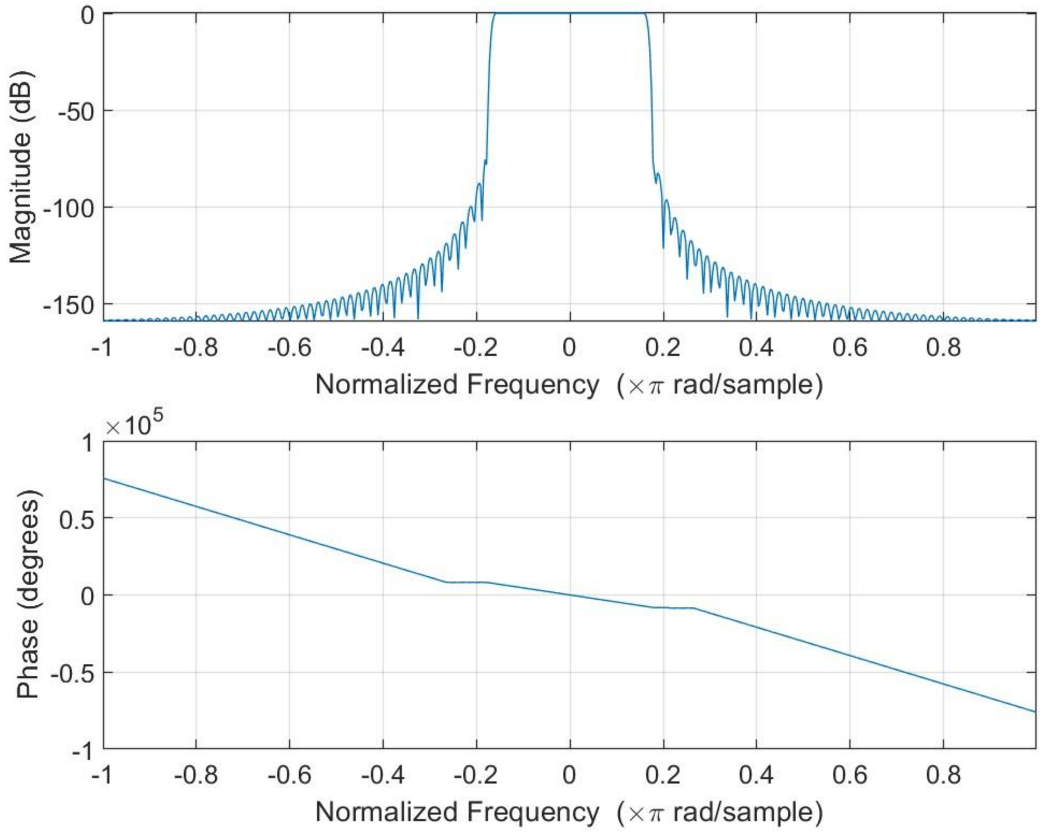
N = 8



N = 64

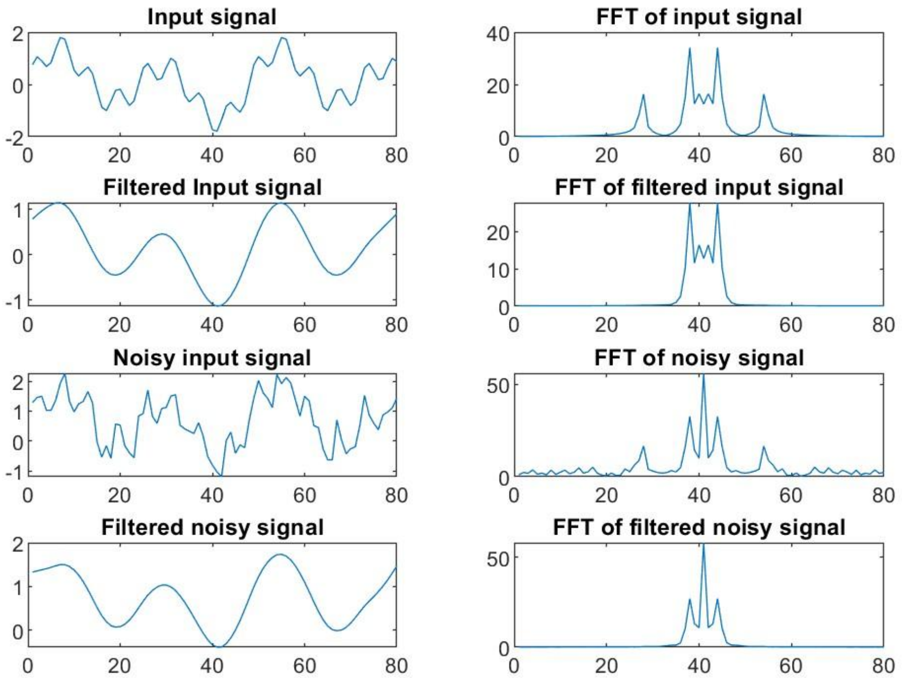


N = 512

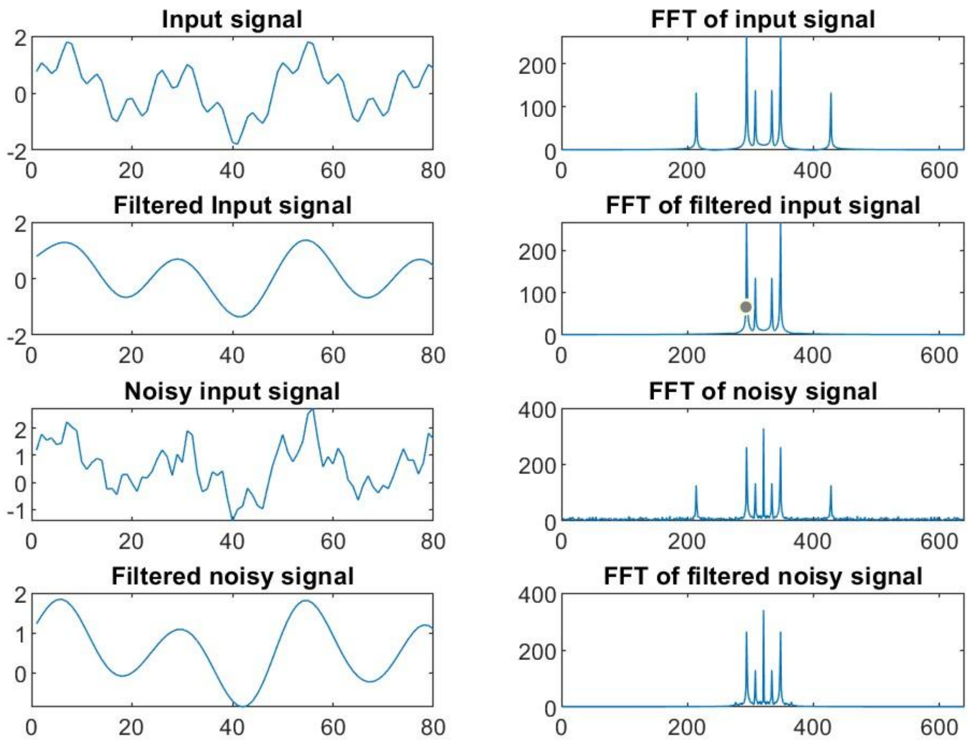

Magnitude (dB)
0
-50
-100
-150
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
Normalized Frequency (×π rad/sample)
×10^5
Phase (degrees)
1
0.5
0
-0.5
-1
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
Normalized Frequency (×π rad/sample)

3. Filter Response to Input signal with and without noise in Time Domain and Frequency Domain:

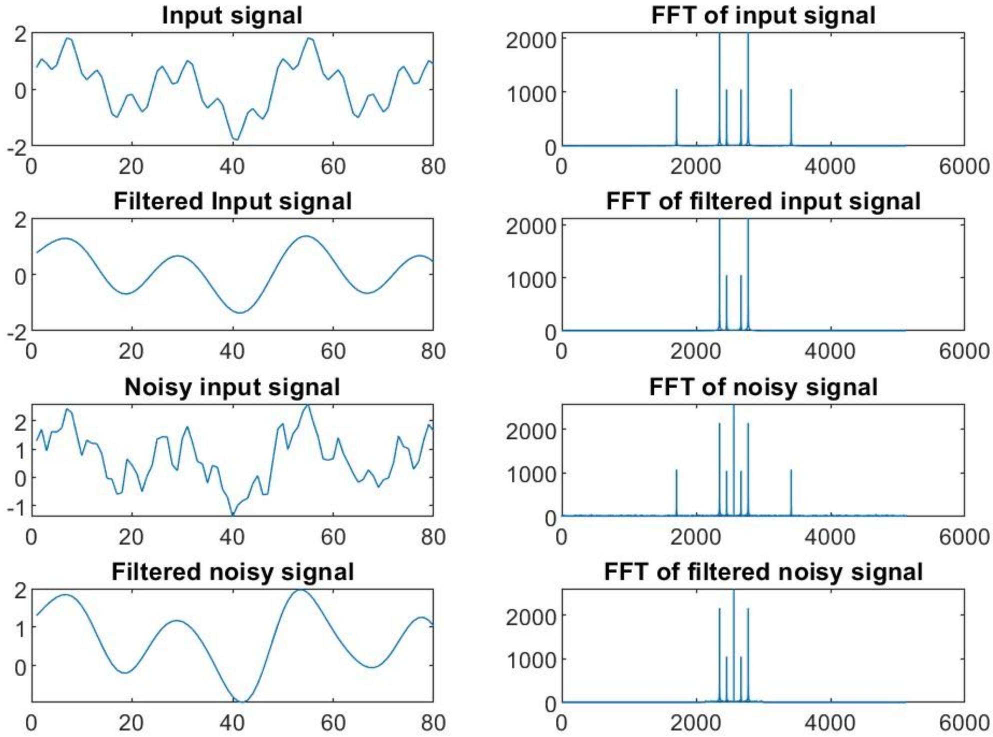
Rectangular Window(N=8)



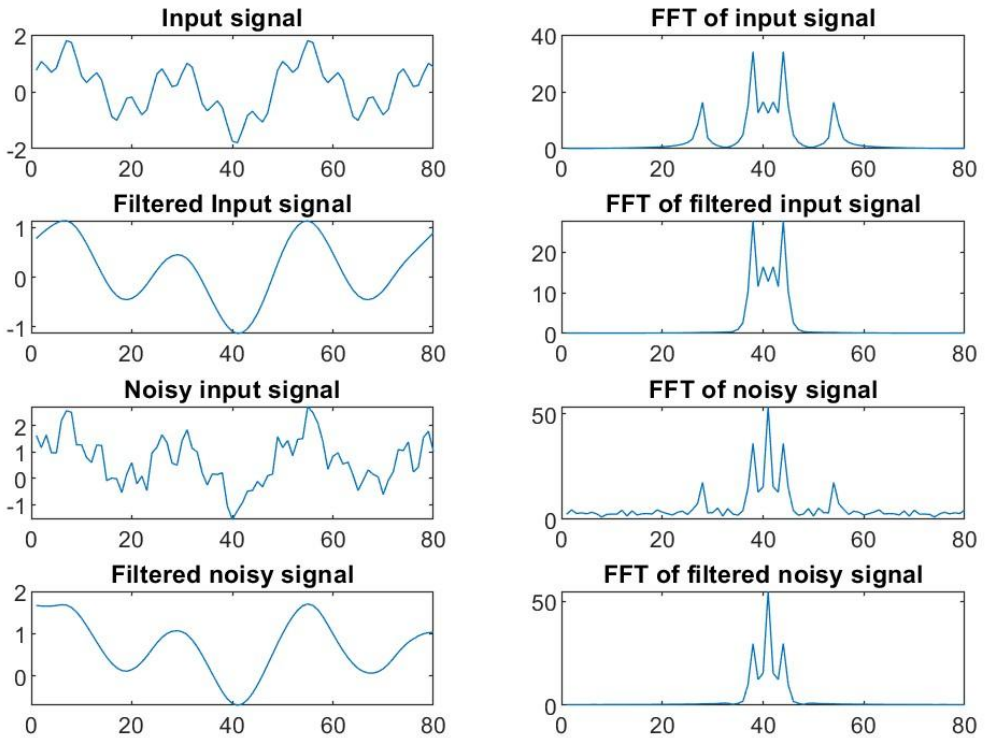
Rectangular Window(N=64)



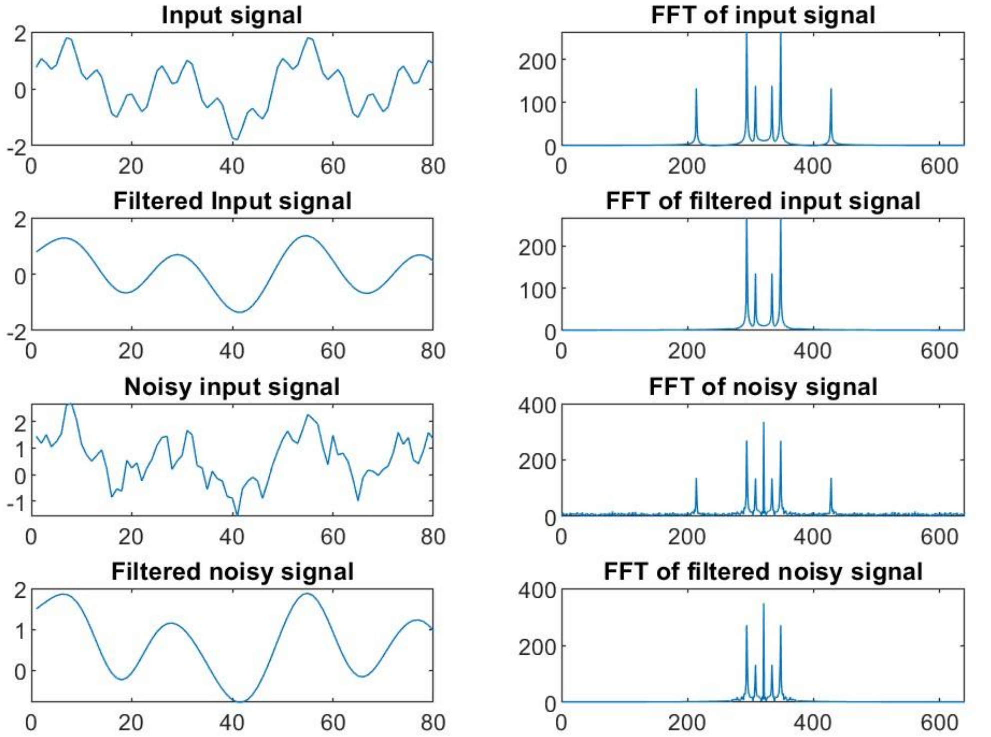
Rectangular Window(N=512)



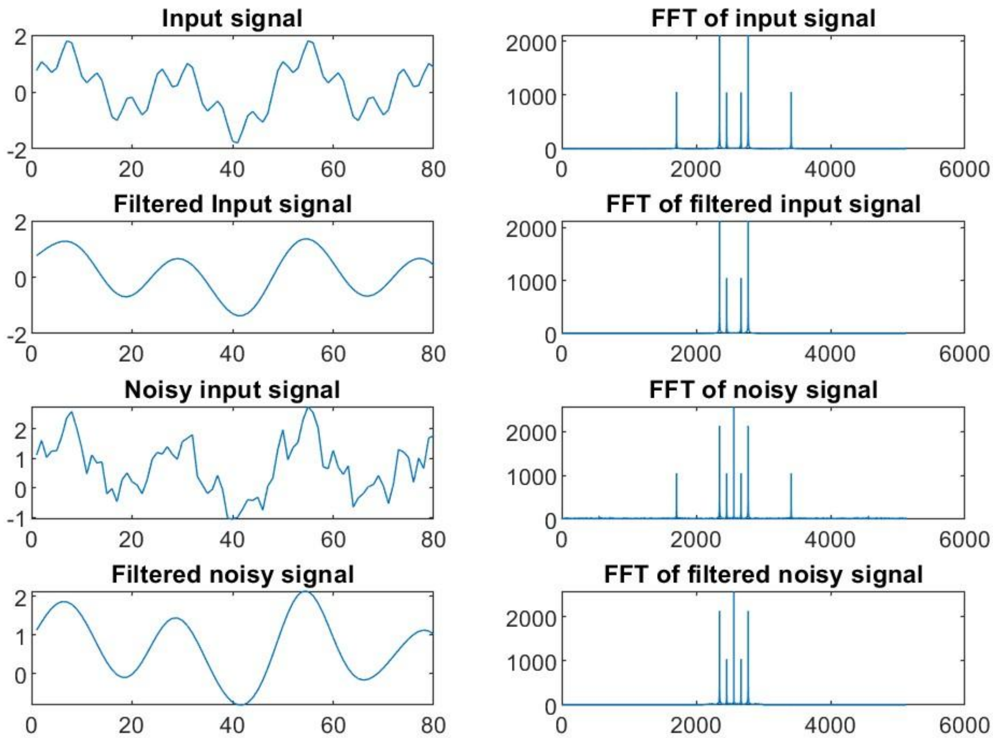
Triangular Window(N=8)

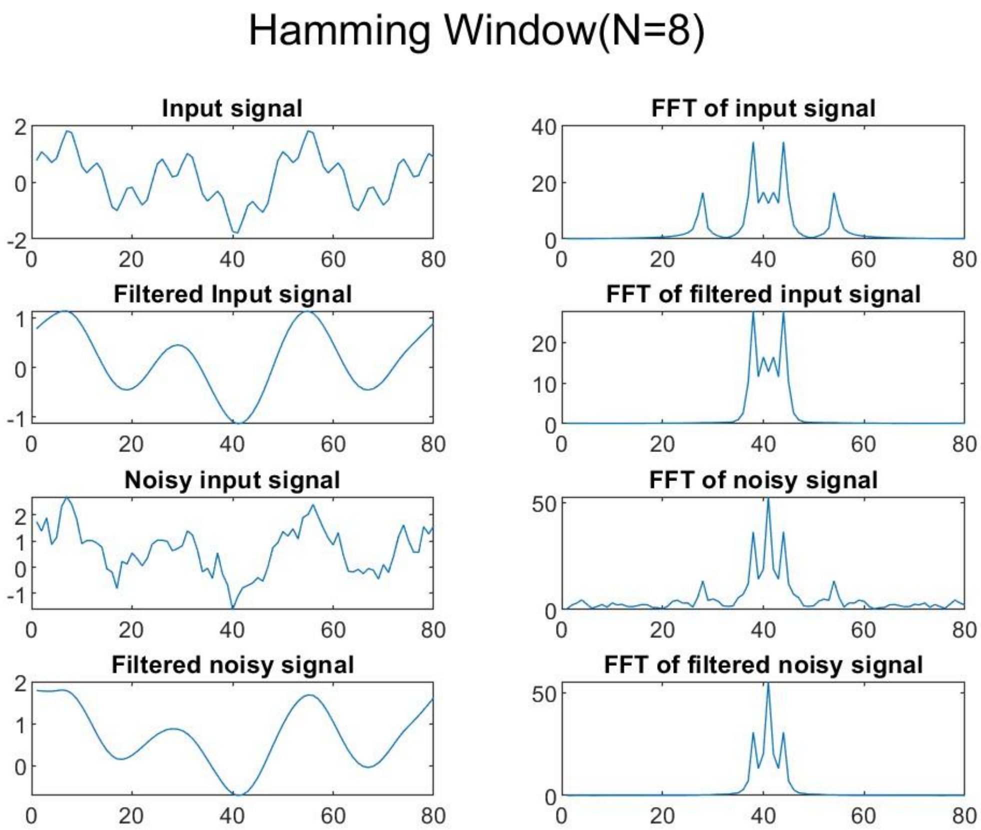


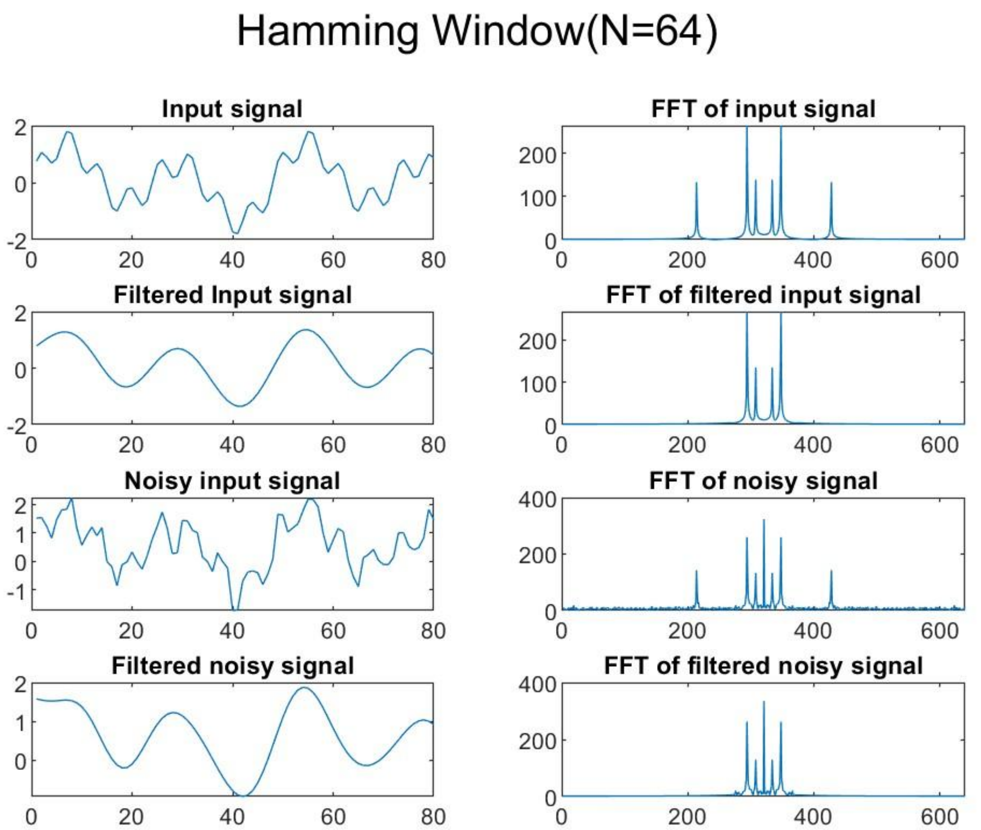
Triangular Window(N=64)



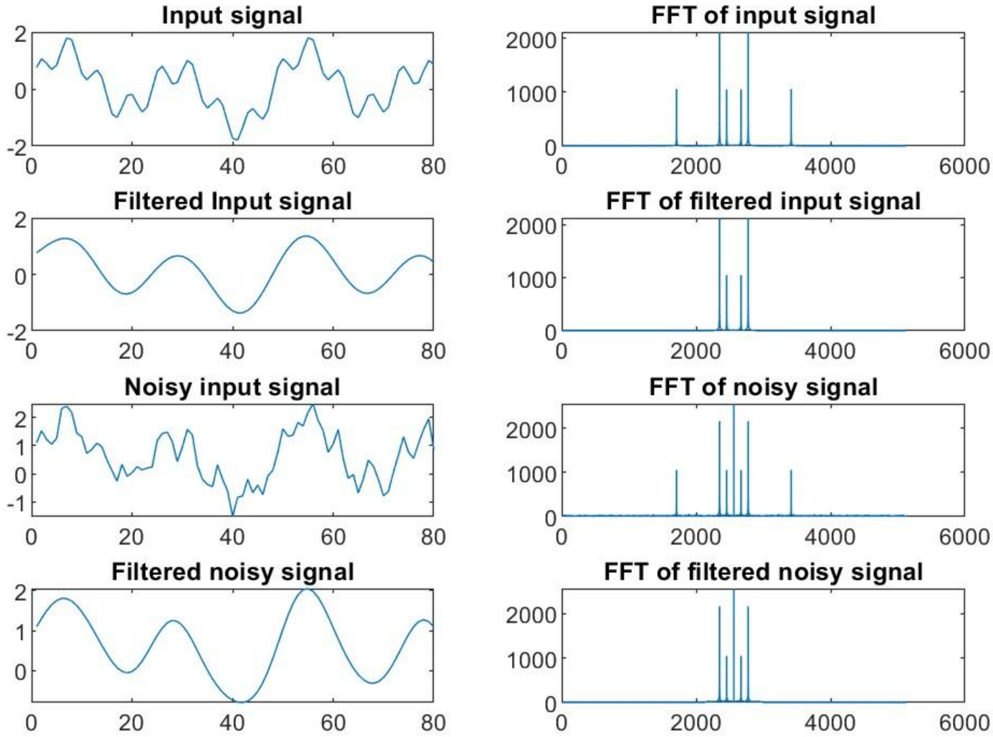
Triangular Window(N=512)



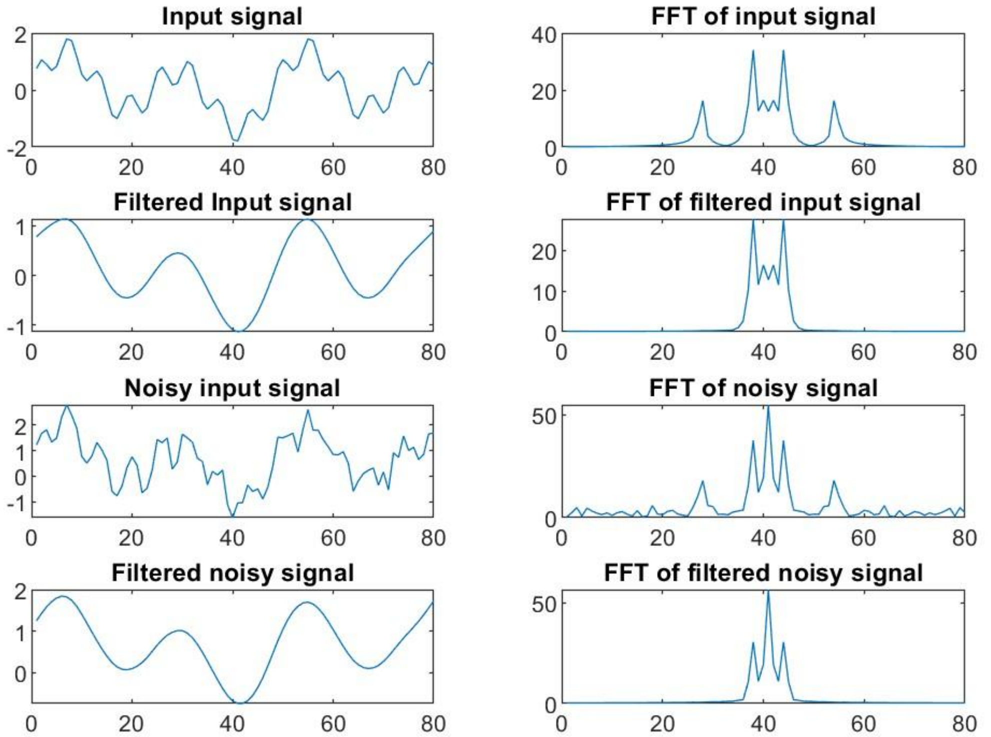

Hamming Window(N=8)
Input signal
FFT of input signal
Filtered Input signal
FFT of filtered input signal
Noisy input signal
FFT of noisy signal
Filtered noisy signal
FFT of filtered noisy signal


Hamming Window(N=64)
Input signal
FFT of input signal
Filtered Input signal
FFT of filtered input signal
Noisy input signal
FFT of noisy signal
Filtered noisy signal
FFT of filtered noisy signal

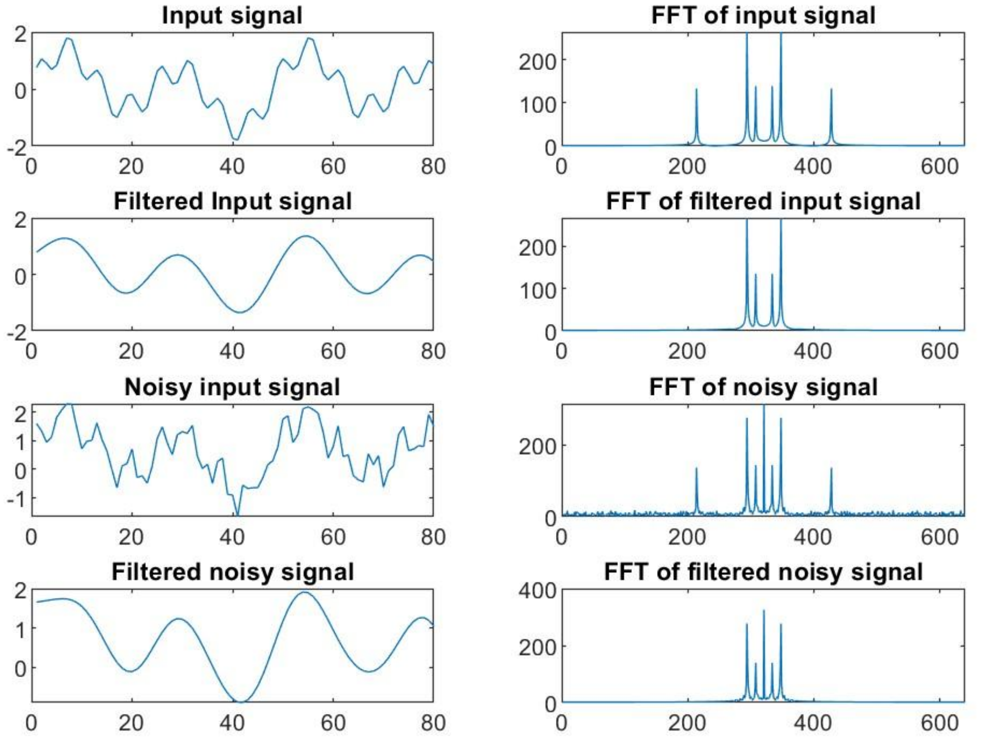
Hamming Window(N=512)



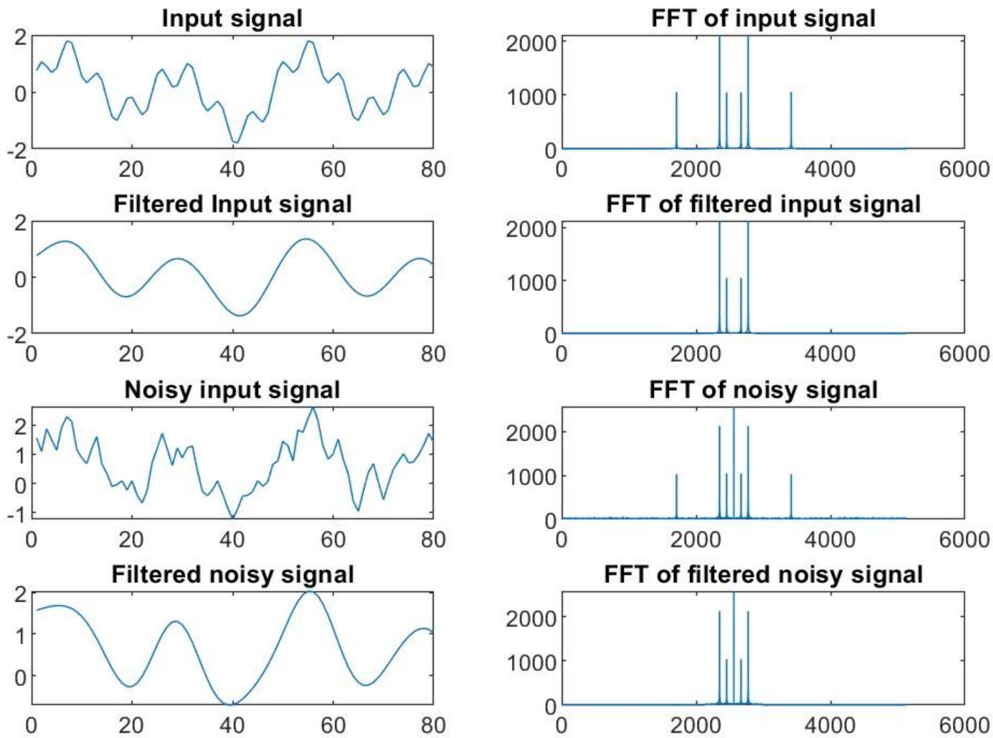
Hanning Window(N=8)



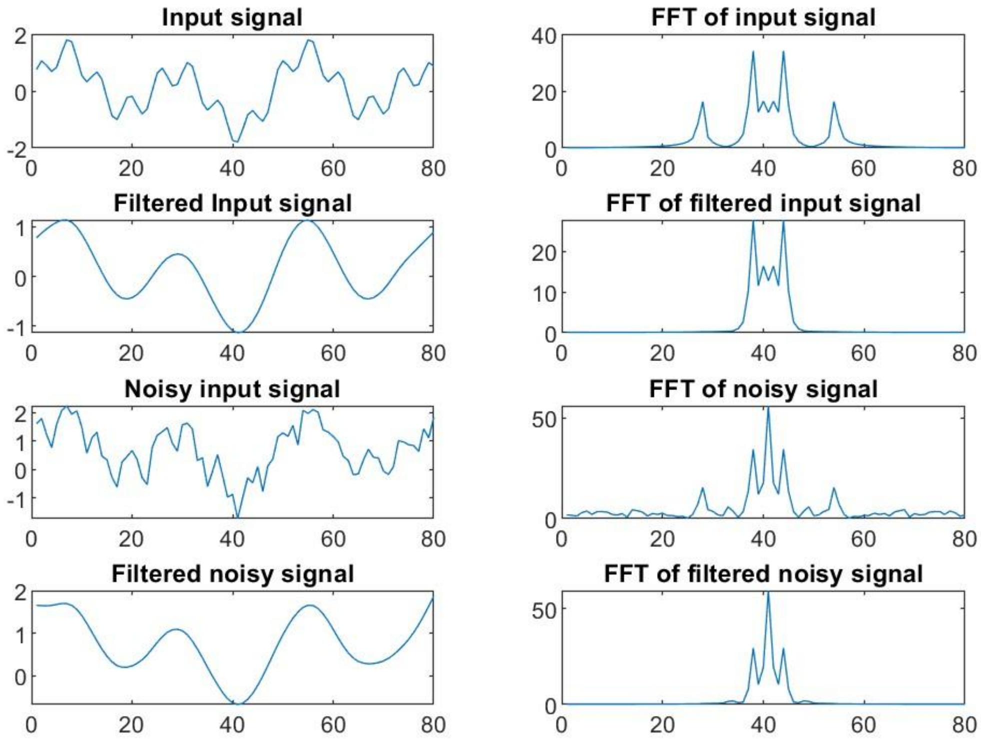
Hanning Window(N=64)



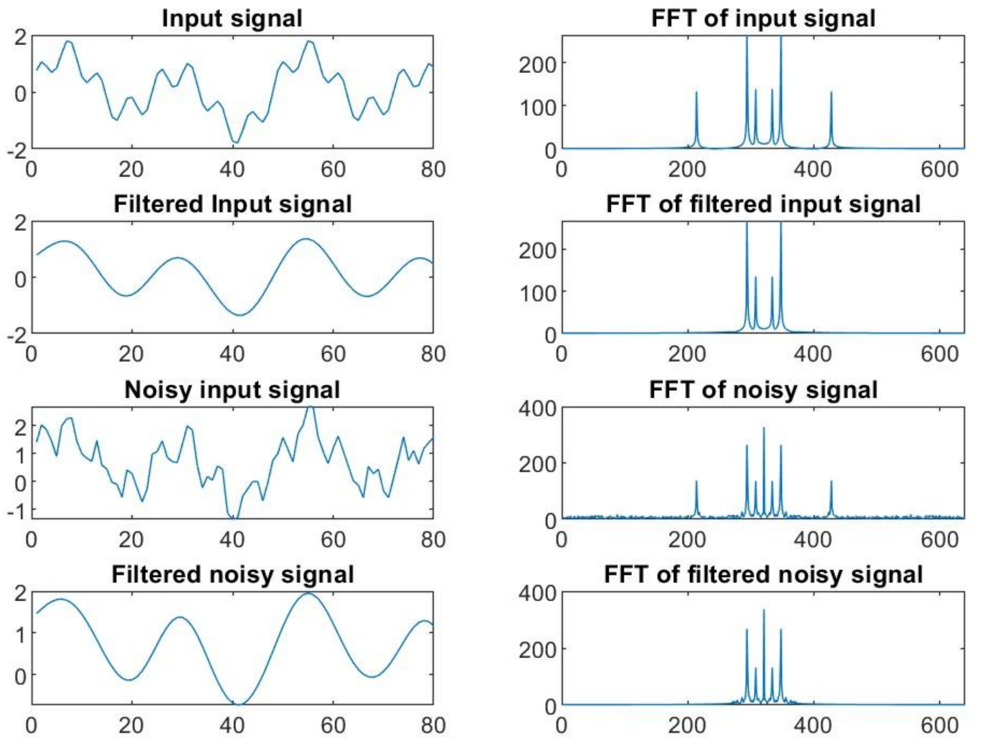
Hanning Window(N=512)



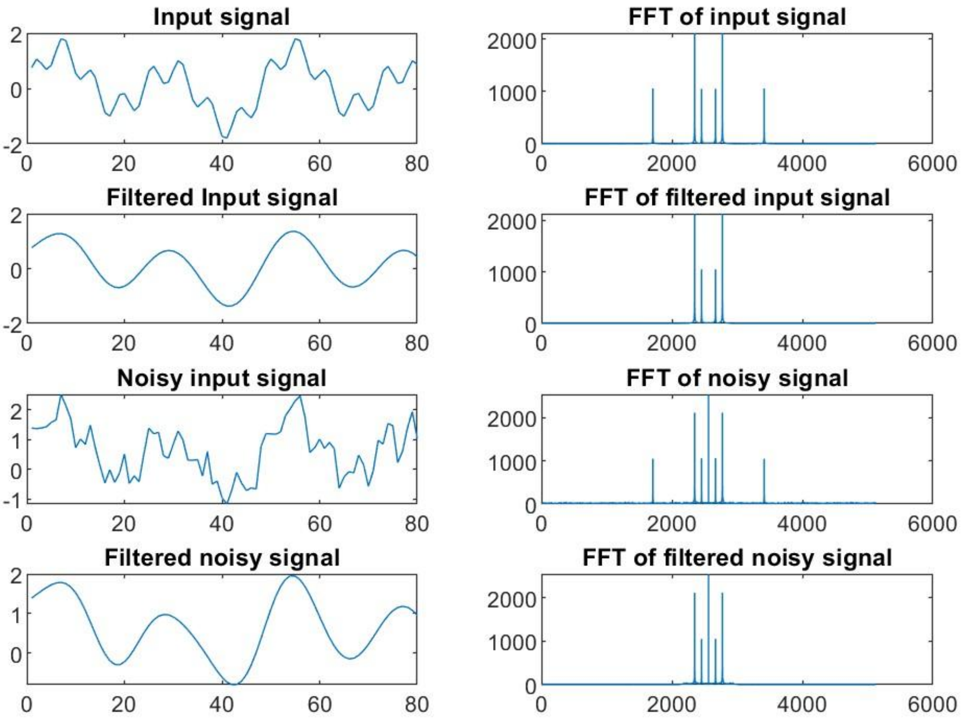
Blackman Window(N=8)



Blackman Window(N=64)



Blackman Window(N=512)



Tables

1.

N	Rectangular Window		
	Transition Width (x **π**)	Peak of first lobe (in dB)	Maximum Stopband Attenuation (in dB)
8	0.1814	-18.0004	-28.5579
64	0.0286	-21.5743	-53.2487
512	0.0078	-36.5601	-70.3879

N	Triangular Window		
	Transition Width (x **π**)	Peak of first lobe (in dB)	Maximum Stopband Attenuation (in dB)
8	0.3915	-26.5102	-45.8185
64	0.0764	-27.4678	-90.7248
512	0.8012	-56.6582	-112.5241

N	Hamming Window		
	Transition Width (x **π**)	Peak of first lobe (in dB)	Maximum Stopband Attenuation (in dB)
8	0.4138	-31.1411	-38.6347
64	0.0668	-44.1448	-93.3159
512	0.0132	-71.3126	-150.4964

N	Hanning Window		
	Transition Width (x **π**)	Peak of first lobe (in dB)	Maximum Stopband Attenuation (in dB)
8	0.4234	-31.9783	-40.3832
64	0.0700	-52.7877	-73.5294
512	0.0140	-59.2835	-91.6131

N	Blackman Window		
	Transition Width (x **π**)	Peak of first lobe (in dB)	Maximum Stopband Attenuation (in dB)
8	0.5528	-43.7402	-40.2915
64	0.1114	-78.0673	-102.6440
512	0.0157	-82.4645	-159.3164

2.

N	Rectangular Window		
	Signal Amplitude	Noise Amplitude	SNR
8	1.6457	0.9797	5.9632
64	2.1369	0.9981	5.0247
512	2.1085	0.9998	5.0831

N	Triangular Window		
	Signal Amplitude	Noise Amplitude	SNR
8	1.7021	0.9789	5.6146
64	2.1363	0.9985	5.2043
512	2.1638	0.9996	5.1778

N	Hamming Window		
	Signal Amplitude	Noise Amplitude	SNR
8	1.7994	0.9859	5.7339
64	2.0728	0.9969	5.3214
512	2.1618	1.0000	5.2117

N	Hanning Window		
	Signal Amplitude	Noise Amplitude	SNR
8	1.8426	0.9957	5.7812
64	2.0350	0.9978	5.5826
512	2.1600	0.9999	5.1076

N	Blackman Window		
	Signal Amplitude	Noise Amplitude	SNR
8	1.8670	0.9943	5.0966
64	2.0210	0.9996	5.3279
512	2.1543	0.9999	5.1980

Discussion:

- It is observed the transition width gets smaller as the value of N is increased. This means that the response of the filter gets sharper and the attenuation of the passband frequencies decreases.

- The magnitude of maximum stopband attenuation and the peak of the first lobe gets suppressed increases as the value of N is increased so the unwanted frequencies get attenuated properly.

- The triangular window has zero ripples in its stopband for N = 512.

- Rectangular window gives the best case for transition width but shows poor attenuation of higher frequency. So, there is a trade-off between transition width and stopband frequency.

- We passed the input signal x = sin((wc/2)*n) + 0.5*sin((wc/4)*n) + 0.5*sin(2*wc*n) where wc is the cutoff frequency of the filter, wc = π/6.

- It is noticed that the frequencies corresponding to "sin(2*wc*n)" term are completely attenuated in the output signal as it lies beyond the cutoff frequency. The same can be better observed in the FFT of the output signal.

- When noise is added to the input signal, a significant amount of unwanted frequencies are observed in the FFT of the input signal.

- It is observed in the FFT of the filtered noisy signal that all the frequencies beyond cutoff frequency are stopped. Therefore in a way, noise also gets filtered to an extent because of the low pass filter.

- In all cases, as the order of filters is increased the SNR ratio improves.

- The amplitude of filtered signal improves as the order of the filter is increased which means that the noise gets filtered better.